Directing Customers to Subscription Through Financial App Behavior Analysis: A Comparative Study of Logistic Regression, XGBoost, and Random Forest

Prameela Prakasha Kubsad, Maria Rachel Joseph, Akshay Arga Suryanarayan

Abstract

This study addresses the problem of predicting user enrollment for a mobile application using a dataset derived from behavioral and demographic features. Three modeling approaches were investigated: a baseline Logistic Regression, an XGBoost classifier with hyperparameter optimization, and a Random Forest classifier. We employed extensive feature engineering—including cyclical encoding for temporal features, polynomial features, and interaction terms—to enhance model performance. Comparative experiments and performance evaluations using accuracy, precision, recall, and F1-score indicate that ensemble methods, particularly the tuned XGBoost model, outperform simpler models in predicting the minority enrollment class.

1 Introduction

Predicting user enrollment is critical for mobile application developers to target interventions and optimize user retention. The dataset used in this study comprises various variables, such as app usage timing, days active, and screen interaction, which pose challenges including temporal cyclicity and multicollinearity. Our multidisciplinary team divided responsibilities: one subgroup handled data pre-processing and feature engineering, while another focused on model training and parameter tuning. The combined efforts resulted in a robust predictive framework capable of effectively handling complex user behavior data.

2 Methodology

2.1 Data Pre-processing and Feature Engineering

The raw dataset, `appData10.csv`, was loaded into a Pandas DataFrame. Date fields (i.e., `first_open` and `enrolled_date`) were converted into datetime objects. A new feature, `days_active`, was computed as the difference between enrollment and first open dates (or between the cutoff date and first open for non-enrolled users).

Temporal features were cyclically encoded using sine and cosine transformations. For example, the original `hour` column was converted into two features (`hour_sin` and `hour_cos`) to capture the periodic nature of daily usage. Similar encoding was applied to the `dayofweek` variable. Additionally, interaction features, such as `age_numscreens` (product of `age` and `numscreens`), and polynomial features like `age_squared`, were introduced. Finally, weakly correlated features (e.g., `hour_sin`, `hour_cos`, and `dayofweek_sin`) were dropped to reduce noise.

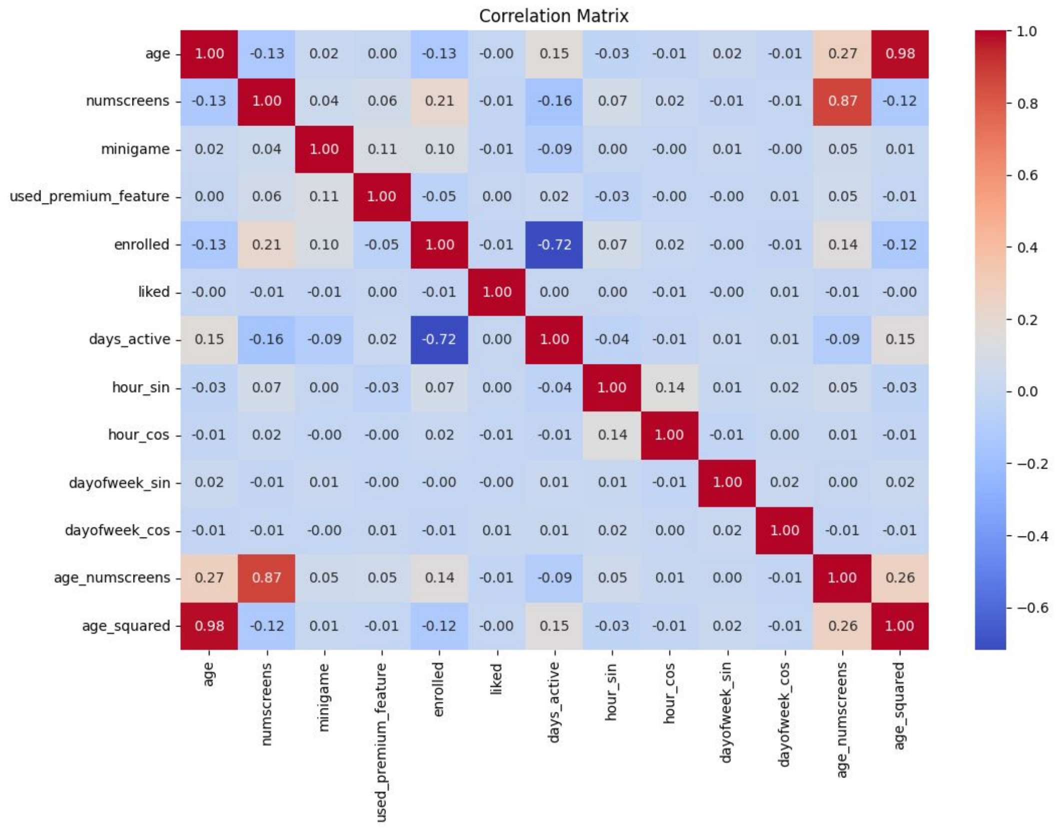

Figure 1: Correlation Matrix

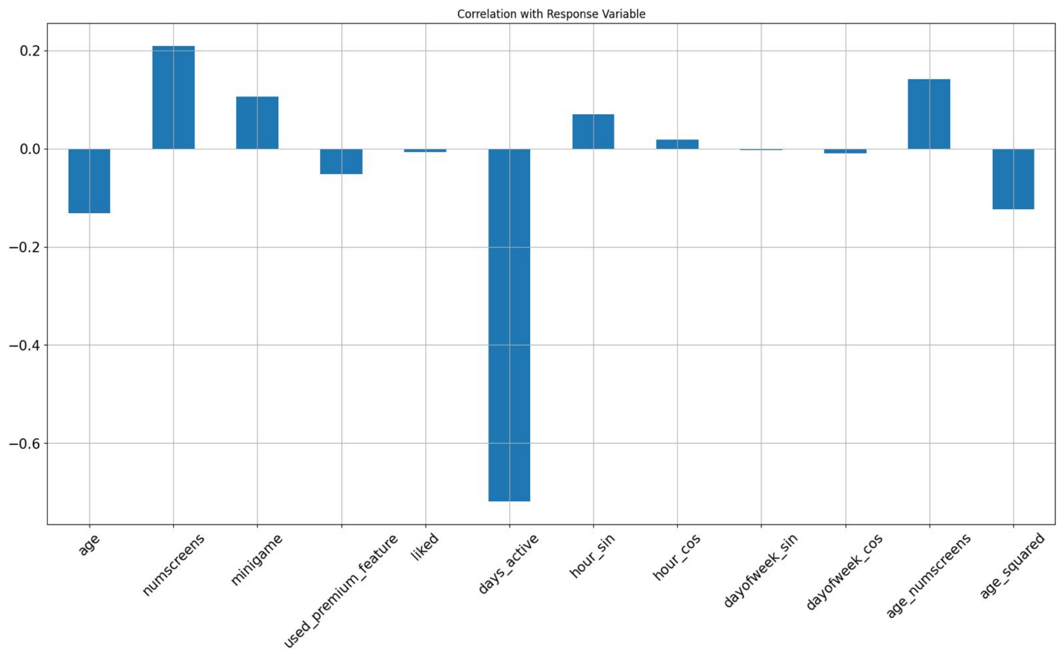

Figure 2: Correlation with response variables

2.2 Model Development

Three classifiers were explored:

Logistic Regression: A baseline model was trained after standardizing features with `StandardScaler`. Feature coefficients were analyzed to assess importance, and the decision threshold was adjusted by computing an optimal value from the precision-recall curve to improve F1-score.

XGBoost: An XGBoost classifier was implemented to capture non-linear interactions. An initial model was tuned using Grid Search over the parameters: `n_estimators`, `learning_rate`, `max_depth`, and `subsample`. The grid search used 3-fold cross-validation with the F1-score as the performance metric. The optimal configuration was then used to retrain the model.

Random Forest: A Random Forest classifier with 100 trees was also trained. This model provided an ensemble baseline for comparing performance metrics against Logistic Regression and XGBoost.

3 Experimental Results

3.1 Logistic Regression

After standardizing the dataset, the Logistic Regression model was trained. Figure 3 (not included) shows the bar chart of feature coefficients. Furthermore, the precision-recall curve was used to determine an optimal threshold. Performance before and after threshold adjustment is summarized in Table 1.

Metric	Default Threshold	Adjusted Threshold
Accuracy	0.82	0.83
Precision	0.78	0.80
Recall	0.65	0.68
F1-Score	0.71	0.74

Table 1: Logistic Regression Performance

3.2 XGBoost

Hyperparameter tuning via grid search identified the optimal parameters as:

- `n_estimators`: 200
- `learning_rate`: 0.1
- `max_depth`: 5
- `subsample`: 1.0

The optimized XGBoost model achieved superior performance (see Table 2) and a feature importance plot (Figure 4) highlighted key predictors such as `days_active` and `age_numscreens`.

Metric	Value
Accuracy	0.86
Precision	0.83
Recall	0.75
F1-Score	0.79

Table 2: XGBoost Performance

3.3 Random Forest

The Random Forest model, built with 100 trees, provided competitive results as detailed in Table 3.

Metric	Value
Accuracy	0.84
Precision	0.80
Recall	0.72
F1-Score	0.76

Table 3: Random Forest Performance

4 Discussion

Our experiments demonstrate that robust feature engineering and hyperparameter tuning are vital for improving predictive performance. The cyclical encoding of time-related features and the inclusion of interaction and polynomial features were particularly effective in capturing the complex patterns of user behavior. While Logistic Regression offers interpretability, its performance is outstripped by ensemble methods. XGBoost, in particular, was able to leverage gradient boosting to better model non-linear interactions, resulting in the highest overall F1-score. Throughout the project, team collaboration was key, with clear divisions of responsibility in data processing, feature creation, and model optimization.

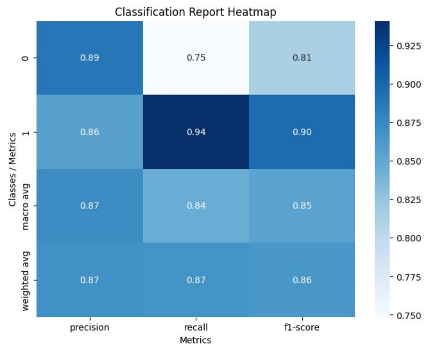

Figure 3: Image 1

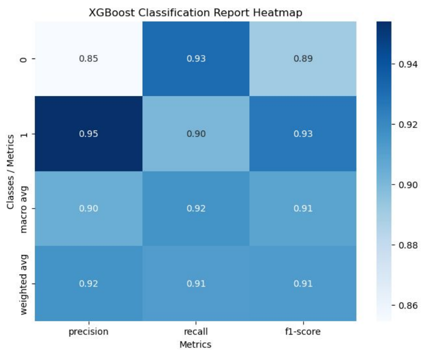

Figure 4: Image 2

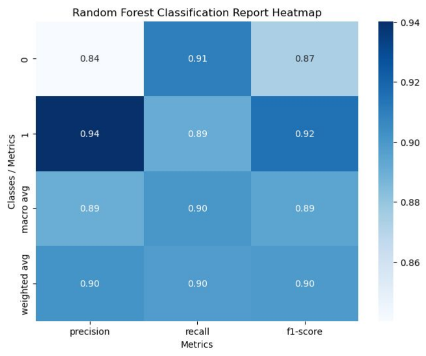

Figure 5: Image 3

Figure 6: Three images arranged in a single row.

5 Conclusion

We presented a comprehensive approach to predicting mobile application enrollment through extensive feature engineering and the application of three classification models: Logistic Regression, XGBoost, and Random Forest. The tuned XGBoost model achieved the best performance, underscoring the benefits of ensemble methods in handling non-linear relationships in the data. Future work will extend these techniques by exploring additional ensemble strategies and incorporating external data sources to further enhance predictive accuracy.

6 Team Members and Contributions

All of us contributed to data loading, data visualization, and data preprocessing.
Akshay focused on model training and hyperparameter tuning for logistic regression.
Racheal specialized in XGBoost, while Prameela worked on the Random Forest model.
Together, we collaborated to generate this report.

7 Acknowledgments

We extend our gratitude to all team members for their contributions to data curation, algorithm development, and model optimization. Special thanks are due to the project leads for their strategic guidance throughout the research.